

Homework 2.3 Question

Blog User Sign-up and Login

Download the handout and unpack it.

You should see three files at the highest level: *blog.py*, *userDAO.py* and *sessionDAO.py*. There is also a *views* directory which contains the templates for the project.

The project roughly follows the model/view/controller paradigm. *userDAO* and *sessionDAO.py* comprise the model. *blog.py* is the controller. The templates comprise the view.

If everything is working properly, you should be able to start the blog by typing:

```
python blog.py
```

Note that this project requires the following python modules be installed on your computer: *cgi*, *hmac*, *datetime*, *json*, *sys*, *string*, *hashlib*, *urllib*, *urllib2*, *random*, *re*, *pymongo*, and *bottle*. If you have python installed at all, you probably already have most of these installed except *pymongo* and *bottle*.

If you have python-setuptools installed, the command "pip" makes this simple. Any other missing packages will show up when *validate.py* is run, and can be installed in a similar fashion. As of this recording, we are still using the beta version of PyMongo so we will install directly from GitHub. Note that this directions are identical to what we taught you within the lessons and you should already have PyMongo 3.x and bottle installed. If not, use the following commands.

```
$ pip install https://github.com/mongodb/mongo-python-  
driver/archive/3.0b1.tar.gz  
$ pip install bottle
```

If you go to <http://localhost:8082> you should see a message, "this is a placeholder for the blog"

Here are some URLs that must work when you are done.

```
http://localhost:8082/signup  
http://localhost:8082/login  
http://localhost:8082/logout
```

When you login or sign-up, the blog will redirect to <http://localhost:8082/welcome> and that must work properly, welcoming the user by username

We have removed two pymongo statements from userDAO.py and marked the area where you need to work with XXX. You should not need to touch any other code. The pymongo statements that you are going to add will add a new user to the database upon sign-up, and validate a login by retrieving the right user document.

The blog stores its data in the blog database in two collections, *users* and *sessions*. Here are two example docs for a username 'erlichson' with password 'fubar'. You can insert these if you like, but you don't need to.

```
> db.users.find()
{ "_id" : "erlichson", "password" :
"d3cadd3699ef6f990d4d53337ed645a3804fac56207d1b0fa44544db1d6c5de,YCRvW"
}
>
> db.sessions.find()
{ "_id" : "wwBfyRDgywSqeFKeQMPqVHPizaWqd1QJ", "username" : "erlichson" }>
```

Once you have the the project working, the following steps should work:

- go to <http://localhost:8082/signup>
- create a user

It should redirect you to the welcome page and say: welcome username, where username is the user you signed up with. Now

- Go to <http://localhost:8082/logout>
- Now login <http://localhost:8082/login>.

Ok, now it's time to validate you got it all working.

There was one additional program that should have been downloaded in the project called *validate.py*.

```
python validate.py
```

For those who are using MongoHQ, MongoLab or want to run the webserver on a different host or port, there are some options to the validation script that can be exposed by running

```
python validate.py -help
```

If you got it right, it will provide a validation code for you to enter into the box below. Enter just the code, no spaces. Note that your blog must be running when you run the validator.